

Nine Questions Scrum Masters and Product Owners Should Be Asking

by Mike Cohn •

18 Comments

Image not readable or empty

Nine Questions Scrum Masters and Product Owners Should Be Asking

Before becoming a Scrum Master, I had worked as the technical lead on a number of teams. Part of my job on those teams was to make decisions. And I think I did it well. Being decisive and assertive is part of my personality.

But those personality traits didn't serve me as well once I became a Scrum Master. I realized that to succeed as a Scrum Master, I needed to shift from making assertions toward asking more questions. Because that wasn't my natural style--and it wasn't what had earned me any success I'd had through that point in my career--I struggled at first.

But with persistence I got better at asking questions. I'd like to share with you some of my favorite questions to ask. Most of these can be asked of a team whether you are a Scrum Master or a product owner.

Two Questions about Estimates

I often need a really rough estimate from a team. I'm not going to hold them to it. (I'm not asking for a commitment. [Estimating and committing](#) are not the same thing.) I really just want a rough estimate. What I find to work really well in this case is to ask:

I'm not looking for an estimate. But if I asked for an estimate, what unit pops into your minds: Hours, days, weeks, months, or years?

Yes, I know those units overlap--many weeks can be more than a month. But getting an estimate from a team like, "Oh, weeks, a few weeks," is often good enough to make a decision, including perhaps the decision to ask the team to formally estimate the work in a more thorough way.

In situations where I do have a formal estimate from a team, another question I'll often ask is:

How confident are you in that estimate?

What you're looking for here is both the degree of confidence and whether team members agree. An estimate in which most people are 90% confident will be more likely to be accurate than one where confidence levels are all over the place and tend to be lower.

Disagreement among team members in their confidence in an estimate may also indicate the team rushed to create the estimate. That may be fine but consider the estimate to be less reliable.

Three Questions About Team Decisions

As a Scrum Master or a product owner, I sometimes want to get a sense of how thoroughly a team has thought through a decision. Here are three questions I often ask:

- *What are three other options you considered before making this decisions?*
- *What's the worst thing that could happen if we pursue this direction?*
- *What has to go right for this to be the best decision?*

You probably don't want to ask all three of these questions. And don't ask the same questions about every decision a team has made.

Also, you aren't asking these questions because you (as a Scrum Master or product owner) have the right to overrule the team's decision. But, you do have the right to understand how confident the team is in a decision and how aligned they are around the decision.

These questions are designed to uncover disagreement. If you ask "What has to go right for this to be the best decision?" and someone says, "Everything!" that may indicate a problem.

Two Questions about Meetings

I really dislike meetings. If I were dropped in a corridor with snakes at one end and a meeting at the other, I don't know which direction I'd run.

So I try to be diligent in keeping meetings--and the number of participants in a meeting--to a minimum. So there are two questions I often ask at the start of a meeting:

- *Do we need everyone who is here now?*
- *Should anyone else be here?*

The first question is designed to see if we can get by without one or two people. I often see agile teams go too far in pursuit of teamwork and collaboration. Team members will feel like they need to be in every meeting, even ones that are completely irrelevant to them. This is the JavaScript developer who attends a meeting on whether the latest release from the database vendor is worth upgrading to.

If people on your team are being overzealous in their meeting attendance, thank them for their commitment to working collaboratively but assure them they don't need to attend every meeting. Establish a team norm that team members should not participate in a meeting if they won't add or receive sufficient value.

(And yes, this can be abused. You'll have to tell some people this does not mean they can opt of every meeting. Ultimately, the team as a whole should have the right to overrule one person's desire not to attend a meeting.)

And, along those lines, the second question above will help identify if someone absent from the meeting should be there. Yes, as much as I hate meetings (I'd probably run toward the snakes), sometimes we need to include more people.

Err on the side of too few meetings and too few people, but some meetings are worth having. And those meetings are made more valuable by having the right participants.

One Question to Ask When Wandering Around

Especially when working as a Scrum Master, I spend a lot of time dropping in on conversations. It's what is traditionally known as *management by wandering around*. For example, if I see a programmer and tester having what seems to be an important conversation, I might walk over and listen in case I can help with anything.

(Obviously don't do this every time they talk or if it looks like a private conversation.)

Sometimes the discussions I hear might be valuable to someone else. For example, perhaps I believe a tech writer should know about whatever the programmer and tester decided. So I'll ask:

- *Does anyone else need to know about this?*

If so, I'll offer to be the one to go share the information if I can. If not, I'll offer to go get the person so they can be included right away.

One Question to Ask During Daily Standups

During a daily scrum, I'll often look at a team's [sprint burndown chart](#) and wonder how they're going to finish everything by the planned end of the sprint. But when I ask this same team if they're going to finish everything, the response is usually that they will.

If I don't agree that their prediction is realistic, I'll look back at the burndown and ask:

- *What do you know that I don't know?*

I might get an answer that one team member hasn't updated their hours in the tool they're using. Or someone will explain that while they're currently behind, they learned a lot and things are just about to speed up. (I find this belief to rarely hold true, but I hear it a lot.)

Asking a team what they know that I don't creates a great opportunity to synchronize assumptions. Perhaps they are assuming things will speed up and I'm not. It's a great question for uncovering different assumptions.

Asking Questions Reveals More than Making Statements

When I first became a Scrum Master and hadn't yet learned the power of questions, I often

missed learning things about my teams and their work. Only after a while did I discover that the best way for me to learn things was to ask a question and then truly listen to the answer.

I hope you'll find these questions as useful as I have.

What Do You Think?

What are some of your favorite questions to ask teams? Please share your thoughts in the comments section below.

Posted: August 29, 2017

Tagged: scrummaster, user stories, estimating

About the Author

As the founder of Mountain Goat Software, Mike Cohn specializes in helping companies adopt and improve their use of agile processes and techniques to build extremely high-performance teams. He is the author of *User Stories Applied for Agile Software Development*, *Agile Estimating and Planning*, and *Succeeding with Agile*. Mike is a founding member of the Agile Alliance and Scrum Alliance. Mike can be reached at hello@mountaingoatsoftware.com. If you want to succeed with agile, you can also [have Mike email you a short tip](#) each week.
